

New Efficient Implicit Time Integration Method for DGTD Applied to Sequential Multidomain and Multiscale Problems

Luis E. Tobon¹, Qiang Ren², Qingtao Sun², Jiefu Chen³, and Qing Huo Liu^{2, *}

Abstract—The discontinuous Galerkin’s (DG) method is an efficient technique for packaging problems. It divides an original computational region into several subdomains, i.e., splits a large linear system into several smaller and balanced matrices. Once the spatial discretization is solved, an optimal time integration method is necessary. For explicit time stepping schemes, the smallest edge length in the entire discretized domain determines the maximal time step interval allowed by the stability criterion, thus they require a large number of time steps for packaging problems. Implicit time stepping schemes are unconditionally stable, thus domains with small structures can use a large time step interval. However, this approach requires inversion of matrices which are generally not positive definite as in explicit schemes for the first-order Maxwell’s equations and thus becomes costly to solve for large problems. This work presents an algorithm that exploits the sequential way in which the subdomains are usually placed for layered structures in packaging problems. Specifically, a reordering of interface and volume unknowns combined with a block LDU (Lower-Diagonal-Upper) decomposition allows improvements in terms of memory cost and time of execution, with respect to previous DGTD implementations.

1. INTRODUCTION

The discontinuous Galerkin time-domain (DGTD) methods are promising in transient analysis of large and multiscale problems [1–9]. Based on the idea of domain decomposition, the DG methods can handle problems too large to be solved by conventional numerical techniques. Basically, the DG methods divide an original computational domain into several well designed subdomains. Within each subdomain, the sizes of the elements are at the same level. In this way, a large system matrix is divided into several smaller and balanced matrices. Thus, once the spatial discretization is defined, an optimal time integration method is crucial.

Time integration in transient multiscale modeling can be very challenging using explicit schemes; small cells needed for capturing fine details (e.g., a multiscale package-to-chip structure in Figure 1) will lead to an extremely small time step Δt , according to the Courant-Friedrichs-Lewy (CFL) condition [10]. Consequently, the large number of calculation in time integration is unaffordable. On the other hand, implicit schemes can surpass the CFL limits because they are unconditionally stable. Thus, subdomains with electrically small structures can use a larger time step interval. This advantage is at the cost of inversion of matrices. Unlike explicit schemes, the matrices in an implicit scheme are not positive definite, and thus usually become more costly to invert. Several works have proposed algorithms to accelerate the implicit time stepping, such as the Block-Thomas (BT) algorithm introduced by [8].

This work presents an algorithm, based on the Crank-Nicholson method [11–15], that exploits the sequential way in which the subdomains are usually placed for layered structures. The coupling between different subdomains leads to a linear block tri-diagonal system. This system is transformed into a block LDU (Lower-Diagonal-Upper) decomposition, resulting in a non-iterative and efficient algorithm. Numerical experiments show the benefits of this new method for sequential systems, in comparison to algorithms based on LU decomposition or the Block-Thomas algorithm [8].

Received 22 November 2014, Accepted 13 February 2015, Scheduled 8 March 2015

* Corresponding author: Qing Huo Liu (qliu@duke.edu).

¹ Pontificia Universidad Javeriana-Cali, Colombia. ² Duke University, USA. ³ Weatherford International, USA.

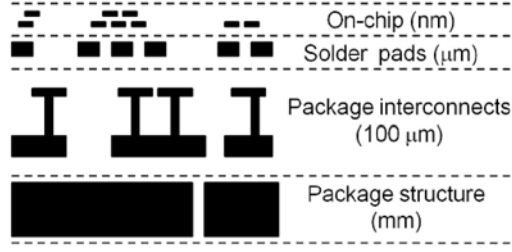


Figure 1. Multiscale package-to-chip structure.

2. GOVERNING EQUATIONS

The DGTD method developed here is based on the first-order Maxwell's equations with electric field \mathbf{E} and magnetic field \mathbf{H} as variables

$$\epsilon \frac{\partial \mathbf{E}}{\partial t} = \nabla \times \mathbf{H} - \sigma_e \mathbf{E} - \mathbf{J}_S \quad (1)$$

$$\mu \frac{\partial \mathbf{H}}{\partial t} = -\nabla \times \mathbf{E} - \sigma_m \mathbf{H} - \mathbf{M}_S \quad (2)$$

where \mathbf{J}_S and \mathbf{M}_S are electric and magnetic current densities; and, ϵ , μ , σ_e and σ_m represent material's permittivity, permeability, electric conductivity, and magnetic conductivity, respectively.

For the spatial discretization, we assume the i th subdomain as the local one and the j th as an adjacent one, and use the curl-conforming basis functions [16–19] $\Phi^{(i)}$ and $\Psi^{(i)}$ to represent \mathbf{E} and \mathbf{H} of the local subdomain, respectively. The discontinuous Galerkin's weak forms of (1) and (2) are

$$\int_{\Omega_i} \Phi^{(i)} \cdot \epsilon \frac{\partial \mathbf{E}}{\partial t} dV = \int_{\Omega_i} \nabla \times \Phi^{(i)} \cdot \mathbf{H} dV - \int_{\Omega_i} \Phi^{(i)} \cdot \sigma_e \mathbf{E} dV - \int_{\Omega_i} \Phi^{(i)} \cdot \mathbf{J}_S dV + \int_{\partial\Omega_i} \Phi^{(i)} \cdot (\mathbf{n} \times \mathbf{H}) dS \quad (3)$$

$$\int_{\Omega_i} \Psi^{(i)} \cdot \mu \frac{\partial \mathbf{H}}{\partial t} dV = - \int_{\Omega_i} \nabla \times \Psi^{(i)} \cdot \mathbf{E} dV - \int_{\Omega_i} \Psi^{(i)} \cdot \sigma_m \mathbf{H} dV - \int_{\Omega_i} \Psi^{(i)} \cdot \mathbf{M}_S dV - \int_{\partial\Omega_i} \Psi^{(i)} \cdot (\mathbf{n} \times \mathbf{E}) dS \quad (4)$$

where Ω_i and $\partial\Omega_i$ denote the volume of the i th subdomain and its boundary, respectively; \mathbf{n} is the outward unit normal vector, and $\mathbf{n} \times \mathbf{E}$ and $\mathbf{n} \times \mathbf{H}$ represent the fluxes on the boundary surface, which are evaluated using the central flux or the Riemann solver [20, 21]. Note that for the i th subdomain, the boundary surface may include the perfect electric conductor (PEC), perfect magnetic conductor (PMC) and shared interface with adjacent subdomains. Only the shared interface contributes to the surface integral terms in (3) and (4), as on PEC or PMC the tangential field components become zero and thus have no contribution.

Assuming N sequential subdomains in total, the discretized system will be a set of matrix equations

$$\mathbf{M}^{(i)} \frac{d\mathbf{u}^{(i)}}{dt} = - \sum_{j=i-1}^{i+1} \mathbf{L}^{(i,j)} \mathbf{u}^{(j)} + \mathbf{q}^{(i)} \quad (5)$$

where $i = 1, \dots, N$; $\mathbf{u}^{(i)}$ is a vector of unknowns; $\mathbf{q}^{(i)}$ is a vector of sources; $\mathbf{M}^{(i)}$ is the mass matrix; and $\mathbf{L}^{(i,j)}$ contains stiffness, damping, and interface matrices, which can be understood as transmission and reflection matrices. We note here that the whole linear system has a block diagonal matrix on the left hand side, and a block tridiagonal matrix on the right, and this characteristic will be exploited to improve the time integration, as explained in the following section.

3. IMPLICIT TIME INTEGRATION METHODS

The Crank-Nicholson method is employed here as the basic implicit scheme to surpass the CFL stability condition of explicit schemes. The solution in the $(n + 1)$ th time step is obtained by solving the below

block tridiagonal system:

$$\begin{bmatrix} \mathbf{A}^{(1)} & \mathbf{B}^{(1,2)} & & & \mathbf{0} \\ \mathbf{B}^{(2,1)} & \ddots & & & \\ & \mathbf{B}^{(i,i-1)} & \mathbf{A}^{(i)} & \mathbf{B}^{(i,i+1)} & \\ & & & \ddots & \\ \mathbf{0} & & & \mathbf{B}^{(N,N-1)} & \mathbf{A}^{(N)} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{n+1}^{(1)} \\ \vdots \\ \mathbf{u}_{n+1}^{(i)} \\ \vdots \\ \mathbf{u}_{n+1}^{(N)} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_n^{(1)} \\ \vdots \\ \mathbf{v}_n^{(i)} \\ \vdots \\ \mathbf{v}_n^{(N)} \end{bmatrix} \quad (6)$$

where

$$\mathbf{A}^{(i)} = \mathbf{M}^{(i)} + \frac{1}{2}\Delta t\mathbf{L}^{(i,i)}$$

$$\mathbf{B}^{(i,j)} = \frac{1}{2}\Delta t\mathbf{L}^{(i,j)}$$

$$\mathbf{v}_n^{(i)} = \begin{bmatrix} \mathbf{D}^{(i,i-1)} & \mathbf{C}^{(i)} & \mathbf{D}^{(i,i+1)} \end{bmatrix} \begin{bmatrix} \mathbf{u}_n^{(i-1)} \\ \mathbf{u}_n^{(i)} \\ \mathbf{u}_n^{(i+1)} \end{bmatrix} + \Delta t\mathbf{q}_{n+\frac{1}{2}}^{(i)} \quad (7)$$

$$\mathbf{C}^{(i)} = \mathbf{M}^{(i)} - \frac{1}{2}\Delta t\mathbf{L}^{(i,i)}$$

$$\mathbf{D}^{(i,j)} = -\frac{1}{2}\Delta t\mathbf{L}^{(i,j)}$$

The Block-Thomas algorithm presented in [8] was designed to accelerate the process of solving (6). It is based on a block LU decomposition, with forward and backward substitutions. This algorithm is very accurate and fast as it is free of iteration. However, the drawback of the method is the dense matrices appearing in the block matrix U. These matrices are memory expensive which limit the size of the EM problems to be solved. We can clarify this point with the following two-subdomain system

$$\begin{bmatrix} \mathbf{A}^1 & \mathbf{B}^{1,2} \\ \mathbf{B}^{2,1} & \mathbf{A}^2 \end{bmatrix} \begin{bmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}^1 & \mathbf{0} \\ \mathbf{B}^{2,1} & \mathbf{A}^2 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{G}^1 \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \end{bmatrix} \quad (8)$$

where the matrix \mathbf{G}^1 is a dense one.

Thus, to solve this issue, an LDU decomposition with a reordering of unknowns is implemented here. In this way, the linear system is lighter than that used in BT. The following is the LDU decomposition for two subdomains:

$$\mathbf{LDU} \begin{bmatrix} \mathbf{u}_V^1 \\ \mathbf{u}_V^2 \\ \mathbf{u}_{S+}^1 \\ \mathbf{u}_{S-}^2 \end{bmatrix} = \begin{bmatrix} \mathbf{v}_V^1 \\ \mathbf{v}_V^2 \\ \mathbf{v}_{S+}^1 \\ \mathbf{v}_{S-}^2 \end{bmatrix} \quad (9)$$

where

$$\mathbf{L} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ (\mathbf{W}^1)^T & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & (\mathbf{W}^2)^T & \mathbf{0} & \mathbf{I} \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{A}_V^{1,1} & 0 & 0 & 0 \\ 0 & \mathbf{A}_V^{2,2} & 0 & 0 \\ 0 & 0 & \mathbf{A}_{S^+}^{1,2} & \mathbf{B}^{1,2} \\ 0 & 0 & \mathbf{B}^{2,1} & \mathbf{A}_{S^-}^{2,1} \end{bmatrix}$$

$$\mathbf{U} = \begin{bmatrix} \mathbf{I} & 0 & \mathbf{W}^1 & 0 \\ 0 & \mathbf{I} & 0 & \mathbf{W}^2 \\ 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \end{bmatrix}$$

\mathbf{W}^i are communication matrices between interface unknowns and volume unknowns of the i th subdomain; \mathbf{u}_V^i are volume unknowns in the i th subdomain; and, $\mathbf{u}_{S^-}^i$ and $\mathbf{u}_{S^+}^i$ are unknowns on the interfaces of the i th subdomain connected with the $(i-1)$ th and $(i+1)$ th neighbor subdomains. As we can see, the solution of this system includes forward and backward substitutions for L and U block matrices. For the block diagonal matrix D , we solve the volume and interface unknowns separately. Furthermore, the subsystem for the interface unknowns also presents a block tridiagonal form, and thus the Block-Thomas algorithm can apply.

4. NUMERICAL RESULTS

The first evaluation of the proposed algorithm is a simple PEC cavity with multiple subdomains, all with the same number of unknowns of 7000. With these models we can verify the performance of the LDU decomposition with respect to the previous implicit BT method and the classic explicit Runge-Kutta method of 4th order (exRK). Three meshes used in this analysis are shown in Figure 2, with four, eight and sixteen subdomains, respectively. Comparisons of the electric field are presented in Figure 3 for the sixteen subdomains case. We can see a perfect agreement between the implicit LDU, BT and explicit Runge-Kutta methods. Note that for the explicit RK method, only one single domain is divided as a reference (1SD exRK) to compare with the DG based LDU method. This figure shows that the new algorithm does not decrease the accuracy of the solution from the BT algorithm and also presents good consistency with the single domain explicit RK method.

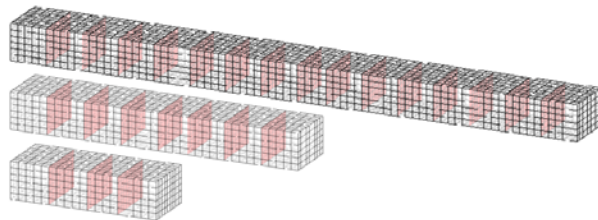


Figure 2. Cavities with four, eight and sixteen subdomains used for verification.

In Figure 4, the memory cost and CPU time for the new algorithm are compared to the BT algorithm and explicit RK for different numbers of subdomains, ranging from 4 to 128. Due to the limitation of computer resource, cases with even larger number of subdomains are not shown, but instead we present the trendlines (the dashed lines) for the three methods based on data fitting. From the figure, we can observe an obvious reduction in memory consumption with LDU (the line with circles), compared to BT (the line with squares). This advantage provides the feasibility of solving large EM problems. With respect to explicit RK (the line with diamonds), however, the LDU method would consume more memory. This is reasonable because the matrix to be inverted for the implicit method is no longer a positive definite one as in the explicit method. Thus it requires more memory. In terms of

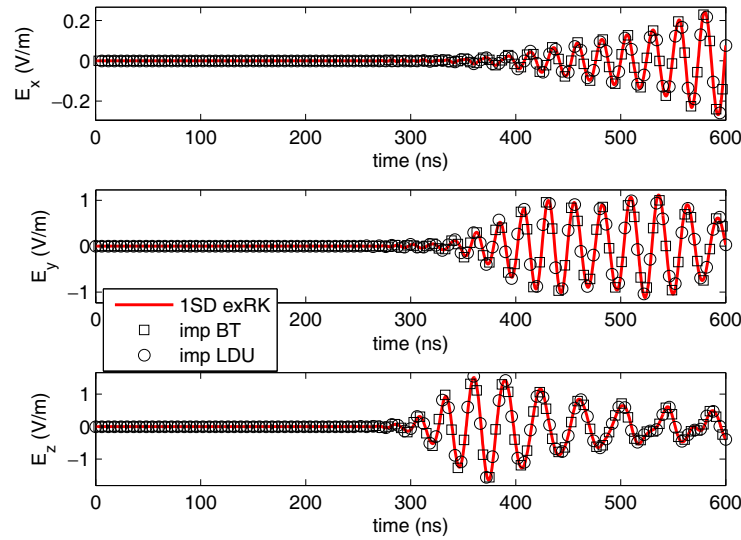


Figure 3. Electric field in x , y , and z direction, case with sixteen subdomains.

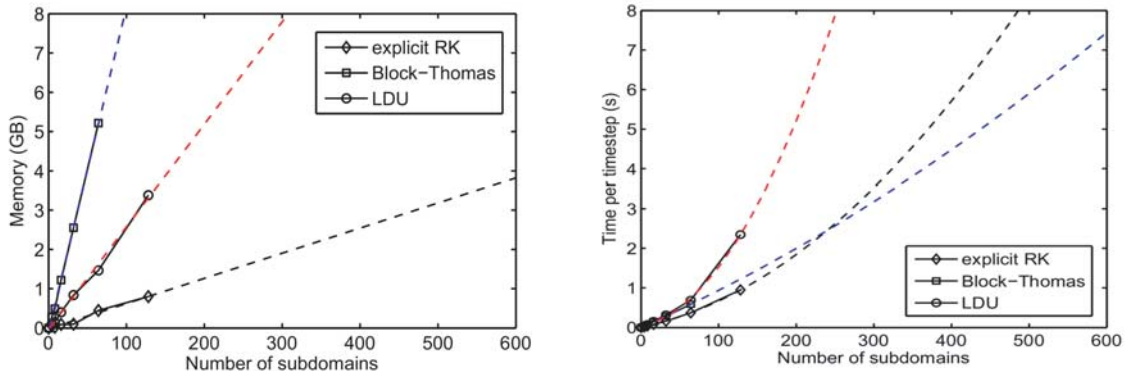


Figure 4. Comparison in memory cost and CPU time for the proposed method.

CPU time per timestep, the LDU method does not show advantage over BT and explicit RK. Especially as the number of subdomains increases, it will take even longer time for each timestep. This drawback, however, can be complemented to some degree by its strengths in other aspects. Compared to BT, although LDU is less efficient in time, it can solve larger problems with more subdomains with the same computer resource. As in this case, LDU can handle up to 128 subdomains, while BT shows difficulty when the subdomain number reaches 64. Compared to explicit RK, even though the LDU method takes longer time for each timestep, it can require much fewer timesteps to complete the simulation, since it is unconditional stable and thereby a large timestep size can be employed. The explicit RK, on the contrary, may require an extremely small timestep size to maintain stability, thus resulting in millions of timesteps, especially for multiscale problems with fine structures.

The next case is designed to verify and evaluate the performance of the LDU method when solving cases with large multiscale factor. The structure is shown in Figure 5, in which the i th layer is scaled as one tenth of the $(i - 1)$ th layer. In this way, the dimension of the layers reduces dramatically as the number of layers increases. The last layer has a subdomain used to evaluate chips with very small structures; in this case, simple conductors with width of $0.15 \mu\text{m}$ are used. As the dimension of the largest layer is 1 cm, the multiscale factor for this case is close to 67 thousand. Finally, eight lumped ports are deployed in each conductor of the bottom layer, with 50Ω impedance. The lumped port number 1 is selected as active, where a first derivative of BHW pulse [22] is generated with a central frequency of 8.5 GHz.

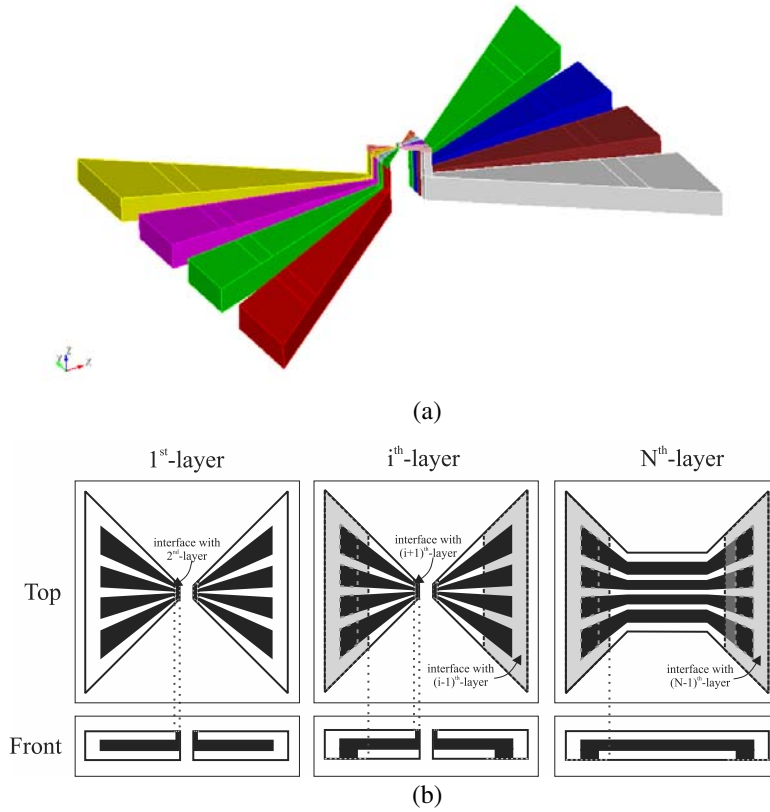


Figure 5. General multiscale package-to-chip structure. (a) 3D view, and (b) top view of the 1st, the i th and the N th layers.

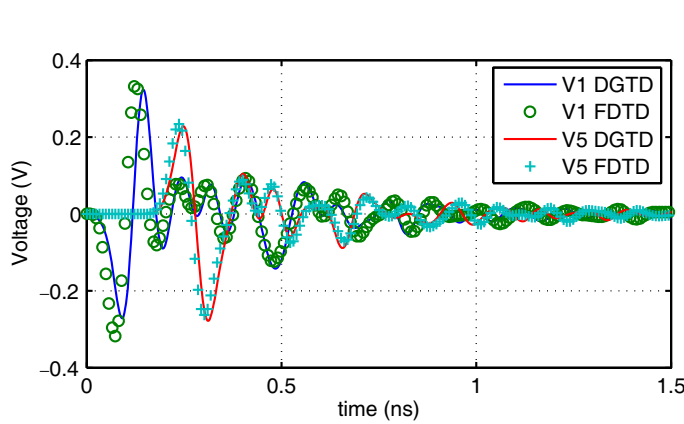


Figure 6. Comparison of transient signals in ports 1 and 5.

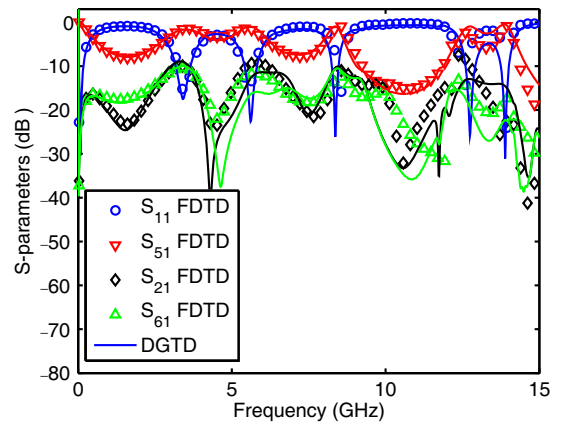


Figure 7. Comparison of S -parameters.

Transient solutions and comparison to FDTD solutions are presented in Figure 6, with good agreement. Similarly, Figure 7 shows the S -parameters S_{11} , S_{21} , S_{51} , and S_{61} , which indicate the good connection between ports 1 and 5, and mutual coupling with ports 2 and 6. The results show some disagreements in high frequencies, and this could be associated with insufficient spatial sampling density in the FDTD method.

Finally, some computational costs are shown in Table 1. The spatial discretization based on DGTD

Table 1. Computational costs.

	Explicit FDTD	Implicit DGTD	Gain
Number of unknowns	3.5 millions	138514	25
Δt	0.36 fs	2 ps	5700
Number of steps for 8 ns	22.8 millions	4000	5700
CPU time per time step (s)	0.0117	0.6	0.02
Total CPU time (s)	265712	2400	110
Memory (MB)	56	1340	0.04

allows a reduction of unknowns in 25 times. The CN based LDU scheme allows larger time step interval with respect to the explicit FDTD, even with very fine structures; in this case, the difference in Δt between the two methods is 5700 times. Finally, an impressive reduction of 110 times of CPU time shows the power of the proposed method.

5. CONCLUSIONS

The discontinuous Galerkin time domain method (DGTD) works accurately and efficiently for highly multiscale EM problems when an optimal time integration method is implemented. Particularly for sequential structure cases, the proposed block Lower-Diagonal-Upper (LDU) decomposition method reduces memory costs and CPU time with respect to the previously implemented Block-Thomas algorithm for the discontinuous Galerkin time domain method. For this reason, the LDU-DGTD method can solve larger cases than traditional single domain methods and the Block-Thomas method. Finally, the LDU-DGTD method also shows important improvements in computational costs with respect to the explicit FDTD, which demonstrates the advantages and potentials of the newly proposed method.

REFERENCES

1. Canouet, N., L. Fezoui, and S. Piperno, "Discontinuous Galerkin time-domain solution of Maxwell's equations on locally-refined nonconforming cartesian grids," *COMPEL: Int. J. for Computation and Maths. in Electrical and Electronic Eng.*, Vol. 24, No. 4, 1381–1401, 2005.
2. Xiao, T. and Q. H. Liu, "Three-dimensional unstructured-grid discontinuous Galerkin method for Maxwell's equations with well-posed perfectly matched layer," *Microw. Opt. Technol. Lett.*, Vol. 46, No. 5, 459–463, 2005.
3. Hesthaven, J. S. and T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, Vol. 54, Springer, 2007.
4. Lee, J.-H. and Q. H. Liu, "A 3-D spectral-element time-domain method for electromagnetic simulation," *IEEE Trans. Microw. Theory Techn.*, Vol. 55, No. 5, 983–991, 2007.
5. Lee, J.-H., J. Chen, and Q. H. Liu, "A 3-D discontinuous spectral element time-domain method for Maxwell's equations," *IEEE Trans. Antennas Propag.*, Vol. 57, No. 9, 2666–2674, 2009.
6. Chen, J. and Q. H. Liu, "A hybrid spectral-element/finite-element method with the implicit-explicit Runge-Kutta time stepping scheme for multiscale computation," *IEEE Intl. Symposium on Antennas and Propagation (APSURSI)*, 1–4, 2010.
7. Chen, J., Q. H. Liu, M. Chai, and J. A. Mix, "A nonspurious 3-D vector discontinuous Galerkin finite-element time-domain method," *IEEE Micro. Wireless Comp. Lett.*, Vol. 20, No. 1, 1–3, 2010.
8. Chen, J., L. Tobon, M. Chai, J. Mix, and Q. H. Liu, "Efficient implicit-explicit time stepping scheme with domain decomposition for multiscale modeling of layered structures," *IEEE Trans. Compon. Packag. Manuf. Technol.*, Vol. 1, No. 9, 1438–1446, 2011.
9. Tobón, L., J. Chen, and Q. H. Liu, "Multilayer microwave filter design using a locally implicit discontinuous Galerkin finite-element time-domain (DG-FETD) method," *2011 IEEE Intl. Symposium on Antennas and Propagation (APSURSI)*, 2972–2975, 2011.

10. Courant, R., K. Friedrichs, and H. Lewy, "On the partial difference equations of mathematical physics," *IBM. J. Res. Dev.*, Vol. 11, No. 2, 215–234, 1967.
11. Sun, G. and C. W. Trueman, "Unconditionally stable Crank-Nicolson scheme for solving two-dimensional Maxwell's equations," *Electron. Lett.*, Vol. 39, No. 7, 595–597, 2003.
12. Sun, G. and C. W. Trueman, "Unconditionally-stable FDTD method based on Crank-Nicolson scheme for solving three-dimensional Maxwell equations," *Electron. Lett.*, Vol. 40, No. 10, 2004.
13. Sun, G. and C. W. Trueman, "Efficient implementations of the Crank-Nicolson scheme for the finite-difference time-domain method," *IEEE Trans. Microw. Theory Techn.*, Vol. 54, No. 5, 2275–2284, 2006.
14. Yang, Y., R. S. Chen, and E. K. N. Yung, "The unconditionally stable Crank-Nicolson FDTD method for three-dimensional Maxwell's equations," *Micro. Opt. Techn. Lett.*, Vol. 48, No. 8, 1619–1622, 2006.
15. Chen, R. S., L. Du, Z. Ye, and Y. Yang, "An efficient algorithm for implementing the Crank-Nicolson scheme in the mixed finite-element time-domain method," *IEEE Trans. Antennas Propag.*, Vol. 57, No. 10, 3216–3222, 2009.
16. Nedelec, J. C., "Mixed finite elements in R^3 ," *Numer. Math.*, Vol. 35, No. 3, 315–341, 1980.
17. Bossavit, A., "Whitney forms: A class of finite elements for three-dimensional computations in electromagnetism," *Proc. Inst. Elect. Eng.*, Vol. 135, No. 8, 493–500, 1988.
18. Lee, J.-H., T. Xiao, and Q. H. Liu, "A 3-D spectral-element method using mixed-order curl conforming vector basis functions for electromagnetic fields," *IEEE Trans. Microw. Theory Techn.*, Vol. 54, No. 1, 437–444, 2006.
19. Chen, J. and Q. H. Liu, "A non-spurious vector spectral element method for Maxwell's equations," *Progress In Electromagnetics Research*, Vol. 96, 205–215, 2009.
20. Shankar, V., A. H. Mohammadian, and W. F. Hall, "A time-domain, finite-volume treatment for the Maxwell equations," *Electromagnetics*, Vol. 10, 127–145, 1990.
21. Mohammadian, A. H., V. Shankar, and W. F. Hall, "Computation of electromagnetic scattering and radiation using a time-domain finite-volume discretization procedure," *Comput. Phys. Commun.*, Vol. 68, 175–196, 1991.
22. Liu, Q. H., "The PSTD algorithm: A time-domain method requiring only two cells per wavelength," *Microw. Opt. Techn. Lett.*, Vol. 15, No. 3, 158–165, 1997.